

AMoRe
a package for
A-utomatic *M*o-lecular *R*_ε-placement

Writeup

Jorge Navaza¹
C.N.R.S. France

¹Acta Crystallographica A50, 157-163. (1994) “*AMoRe* : an Automated Package for Molecular Replacement”.

JOB

JOB creates default inputs and a script with a tentative protocol to solve the molecular replacement problem.

Calling command: `./e/job dato`

Input: `./i/dato.i3` = input described below.

`./d/data.d` = main *AMoRE* input described below.

Output: `./job` = script with protocol for the actual molecular replacement problem.

`./i/sort.i1` = input for SORTING.

`./i/patt.i1` = input for PATTING.

`./i/tabl.i1` = input for TABLING.

`./i/oicrd.i2` = input for OIC (ROTING mode).

`./i/oicto.i2` = input for OIC (TRAINING mode, one-body).

`./i/oictn.i2` = input for OIC (TRAINING mode, n-body).

`./i/oicfd.i2` = input for OIC (FITING mode).

`./i/oicpr.i2` = input for OIC (PROPRE mode).

Input: file ./d/data.d

```
* D44HEL **
99.7 167.3 84.7 90. 90. 90.
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95. 0.
15. 3.5
2 2 2
```

Description

- 1) Title (format A80).
 - 2) Cell.
 - 3) Symmetry operations (lower case), finishing '* end'.
 - 4) NORT
Code to define an orthogonal reference frame.
 - 5) PERC BADD
PERC = uses only the PERC % highest F^{obs} .
BADD = B-factor added to F^{obs} (e.g. -5 to sharpen data).
 - 6) DMAX DMIN
Resolution limits used for the molecular replacement problem (in Å).
 - 7) NTYP MOL1 ... MOLn
NTYP = number of different models.
MOL{#} = number of molecules of model-type {#} in the a.u.;
{#} = 1,NTYP.
-

Input: file ./i/dato.i3

```
job +*+*+*+*+*+*+*+*+*
xyz
1.  2  10  0.5  2.5
c-o  50  0.3  30
p-t  10  0.5  30
10  20
20.
```

Description

- 1) Keyword (format A4) = 'job '.
 - 2) AKEY (format A5)
Keyword defining mode: $\left| \begin{array}{l} \text{'xyz' reads coordinates;} \\ \text{'map' reads electron density map.} \end{array} \right.$
 - 3) RATE LMINs LMINf CUTR STEP
RATE = defines the integration radius as $\text{RATE} \times \text{Molrad}$, where
Molrad = radius of the smallest sphere, with origin at CoM,
containing the whole molecule.
 $\left. \begin{array}{l} \text{LMINs} \\ \text{LMINf} \end{array} \right| = \left| \begin{array}{l} \text{several rotation functions are calculated, where the spherical-} \\ \text{harmonics expansions begin with } \text{LMINs} \leq \ell \leq \text{LMINf}; \text{ the} \\ \ell\text{-expansion controls the angular resolution.} \end{array} \right.$
CUTR = cutoff in rotation function output; first selects all peaks
above $\text{CUTR} \times \text{maximum-peak-height}$.
STEP = step size for ϕ , θ and ψ (in degrees).
 - 4) TKEY NUMR CUTT NPIC (format A5,*)
 $\text{TKEY} = \left| \begin{array}{l} \text{'c-o' computes centered-overlap;} \\ \text{'p-t' computes phased-translation ('p-t-f' when phases} \\ \text{are available);} \\ \text{'h-l' computes Harada-Lifchitz translation function;} \\ \text{'c-c' computes correlation-coefficient.} \end{array} \right.$
NUMR = selects up to a maximum of NUMR orientations for input
to one-body translations.
CUTT = cutoff in fast translation function output; first selects all
peaks above $\text{CUTT} \times \text{maximum-peak-height}$ of 'c-o', 'p-t',
'h-l' or 'c-c' fast translation function.
NPIC = number of peaks to output of translation function. The
program computes correlations and R-factors.
 - 5) Same as previous card, but for n-body translations (it must be present,
even if not used).
 - 6) NUMT NITE
NUMT = selects up to a maximum of NUMT positions to refine.
NITE = number of iterations in the least-squares procedure.
 - 7) CUTD
Cutoff to eliminate positions with CoM-CoM distance less than CUTD Å.
-

Example

The data.d file corresponds to a crystal with two molecules of the complex Fab-Lysozyme in the asymmetric unit, and two search models (Fab = 1 , Lysozyme = 2). JOB proposes the following protocol:

```
# amore
./e/sorting
  set k=1
  while (${k} <= 2)
./e/tabling ${k}
  set k='expr ${k} + 1'
  end
  set k=1
  while (${k} <= 2)
./e/oic_rotating oicrd ${k} o${k}r
  set k='expr ${k} + 1'
  end
cat ./o/o*r.s > ./o/or1.s
./e/oic_training oicto or1 ot1
  set k=1
  while (${k} < 4)
  set m='expr ${k} + 1'
./e/oic_training oictn ot${k} ot${m}
  set k='expr ${k} + 1'
  end
./e/oic_fitting oicfd ot4 of4
./e/mr2ic of4 mr2ic
```

Note that all the one-body rotation outputs (./o/o{#}r.s) have been concatenated into the single file ./o/or1.s . However, since we may expect the Fab positions to be determined more easily than the Lysozymes, we may execute the programs in a different order:

```

# amore
./e/sorting
  set k=1
  while ({k} <= 2)
./e/tabling {k}
  set k='expr {k} + 1'
  end
  set k=1
  while ({k} <= 2)
./e/oic_rotating oicrd {k} o{k}r
  set k='expr {k} + 1'
  end
./e/oic_traing oicto o1r ot1
./e/oic_traing oictn ot1 ot2
./e/oic_fiting oicfd ot2 of2
./e/oic_traing oictn o2r of2 ot3
./e/oic_traing oictn ot3 ot4
./e/oic_fiting oicfd ot4 of4
./e/mr2ic of4 mr2ic

```

Note that fast rigid-body refinement is performed on the 2-body translation output.

SORTING

SORTING packs and sorts H,K,L,F^{obs} for use in later programs. The packing is cell and space-group dependent.

Calling command: ./e/sorting

Input: `./i/sort.i1` = input described below.

./d/hkl.d = ascii file of H,K,L,F^{obs}.

Output: `./f/xudi` = binary file of packed and sorted H, K, L, F^{obs} .

`./o/sort.s` = SORTING output.

Input: file ./i/sort.i1

```

sorting +*+*+*+*+*+*+*+*+*
1 10 9 :logical units
* D44HEL **
99.7 167.3 84.7 90. 90. 90.
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
15.00 3.50
1.00

```


Description

- 1) Keyword (format A7) = 'sorting'.
 - 2) LUN1 LUN2 LUN3
Logical units (see script ./e/sorting):
 - LUN1 = input ascii file of H,K,L,F^{obs} (./d/hkl.d).
 - LUN2 = output binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
 - LUN3 = SORTING output (./o/sort.s).
 - 3) Title (format A80).
 - 4) Cell.
 - 5) Symmetry operations (lower case), finishing '* end'.
 - 6) DMAX DMIN
Resolution limits. Performs statistics for data within this resolution range.
 - 7) SCAL
Dividing scale factor.
-

PATTING

PATTING calculates the Patterson function using the expression

$$P(\mathbf{r}) = \frac{1}{V} \sum_{\mathbf{H}} |F_{\mathbf{H}}|^2 e^{-2\pi i \mathbf{H} \mathbf{r}}$$

Calling command: `./e/patting`

Input: `./i/patt.il` = input described below.

`./f/xudi` = binary file of packed and sorted H, K, L, F^{obs} .

Output: `./o/patt.s` = PATTING output.

To recover the Patterson function map, erase in script `./e/patting` the symbol “#” in line “`#mv $AMOREF/fort.91 patmap`”. `./patmap` is a binary file with several registers:

```
real patmap(nx,ny,nz)
write(file) nx,ny,nz,a,b,c,alpha,beta,gamma
do iz=1,nz
write(file) ((patmap(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```

where:

`nx,ny,nz` = number of sampling points in each direction.
`a,b,c,alpha,beta,gamma` = cell parameters (in Å and degrees).

Description

- 1) Keyword (format A7) = 'patting'.
 - 2) Printing options.
 - 3) LUN1 LUN2
Logical units (see script ./e/patting):
LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
LUN2 = PATTING output (./o/patt.s).
 - 4) Title (format A80).
 - 5) Cell.
 - 6) Symmetry operations (lower case), finishing '* end'.
 - 7) PERC BADD
PERC = uses only the PERC % highest F^{obs}.
BADD = B-factor added to F^{obs} (e.g. -5 to sharpen data).
 - 8) DMAX DMIN SHAR
DMAX,DMIN = data resolution limits.
SHAR = Shannon rate (the greater SHAR, the finer the mesh).
Defines the crystal cell sampling as
$$NX \approx 2 \times a / (DMIN/SHAR)$$
$$NY \approx 2 \times b / (DMIN/SHAR)$$
$$NZ \approx 2 \times c / (DMIN/SHAR)$$
 - 9) CUTP NPIC DELT
CUTP = cutoff in Patterson function; selects all peaks above
CUTP \times maximum-peak-height.
NPIC = maximum number of peaks to output of Patterson function.
DELT = used in peak-search to avoid spurious peaks.
-

TABLING

TABLING calculates the array of molecular scattering factors corresponding to the model coordinates. Puts the model in a small model-box: first translates the coordinates so that the center of mass is at the origin, and rotates the coordinates so that the model's principal axes of inertia are parallel to the model-box axes. The model-box is put in a big cell in order to sample the model Fourier transform finely, to allow structure factors and gradients of the rotating model to be accurately interpolated.

For given orientation \mathbf{R} and translation \mathbf{T} of the search model, the calculated structure factors are expressed in terms of the model scattering factors – f_m – by

$$F_{\mathbf{H}}^{cal(m)}(\mathbf{R}, \mathbf{T}) = \sum_{g=1}^G f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}\mathbf{O}_m) e^{2\pi i\mathbf{H}(\mathbf{M}_g\mathbf{T}+\mathbf{t}_g)}$$

Calling command: `./e/tabling {#}`

Input:	<code>./i/tabl.i1</code>	=	input described below.
	<code>\$AMORE/factor.d</code>	=	ascii file of atomic form-factors.
	<code>./d/xyz{#}.d</code>	=	ascii file of coordinates.
	<code>./d/map{#}.d</code>	=	binary file of electron density.
Output:	<code>./f/tabl{#}</code>	=	binary file of molecular scattering factors.
	<code>./o/tabl{#}.s</code>	=	TABLING output.

Input: file ./i/tabl.i1

tabling +*+*+*+*+*+*+*+*+*+*

3 1 0 1 :printing

1 2 10 9 :logical units

molecular scattering factors

xyz

0 0.0

3.5

3.50 2.00 1 60.5

Description

- 1) Keyword (format A7) = 'tabling'.
 - 2) Printing options.
 - 3) LUN1 LUN2 LUN3 LUN4
Logical units (see script ./e/tabling):
LUN1 = input ascii file of coordinates - ./d/xyz{#}.d
LUN2 = input binary file of electron density - ./d/map{#}.d
LUN3 = output binary file of molecular scattering factors - ./f/tabl{#}
LUN4 = TABLING output - ./o/tabl{#}.s
 - 4) Title (format A80).
The procedure uses "sed" to add "model {#} -" to the title.
 - 5) AKEY (format A5)
Keyword defining mode: $\left\{ \begin{array}{ll} \text{'xyz' ,} & \text{reads coordinates;} \\ \text{'map' ,} & \text{reads electron density map.} \end{array} \right.$
 - 6) BFLG BREP
B-factor flags:
BFLG = 1, replace all atomic B-factors by BREP.
BREP = to set all atomic B-factors to BREP.
 - 7) SBOX
Defines the model-cell as $\text{SBOX} \times \text{model-box}$.
 - 8) DMIN SHAR TFLG BADD
DMIN = resolution limit of generated Fourier coefficients. In fact program generates past this point to allow for interpolation at the frontier.
SHAR = Shannon rate (the greater SHAR, the finer the mesh).
Defines the model-cell sampling as
$$\text{NX} \approx 2 \times \tilde{a} / (\text{DMIN}/\text{SHAR})$$
$$\text{NY} \approx 2 \times \tilde{b} / (\text{DMIN}/\text{SHAR})$$
$$\text{NZ} \approx 2 \times \tilde{c} / (\text{DMIN}/\text{SHAR})$$
where $\tilde{a}, \tilde{b}, \tilde{c}$ are the model-cell parameters.
TFLG = 0, use optimal values for SHAR and BADD.
BADD = "B-factor" to smear the atomic electron density. It is added in direct space (when reading atomic coordinates) and subtracted in reciprocal space (in all cases).
-

Maps

If the search model is an electron density (e.g. a low resolution envelop), TABLING translates it as previously described, but does not rotate it. The input must be a binary file – ./d/map{#}.d – with several registers:

```
real rho(mx,my,mz)
write(file) mx,my,mz,xlw,ylw,zlw,xup,yup,zup,alpha,beta,gamma
do iz=1,mz
write(file) ((rho(ix,iy,iz),ix=1,mx),iy=1,my)
enddo
```

where:

mx,my,mz	=	number of sampling points in each direction.
xlw,ylw,zlw,xup,yup,zup	=	lower and upper limits (in Å) of the model-box.
alpha,beta,gamma	=	model-box angles.

The model-box edges are thus

$$\begin{aligned}\tilde{a} &= xup - xlw \\ \tilde{b} &= yup - ylw \\ \tilde{c} &= zup - zlw\end{aligned}$$

This should be, as far as possible, the smallest box containing the model density. The sampling must be sensibly equal to

$$\begin{aligned}mx &\approx 2 \times \tilde{a} / (DMIN/SHAR) \\ my &\approx 2 \times \tilde{b} / (DMIN/SHAR) \\ mz &\approx 2 \times \tilde{c} / (DMIN/SHAR)\end{aligned}$$

Fourier Coefficients

It is also possible to provide the output of TABLING without executing the program. It must be a binary file – `./f/tabl{#}` – with two registers:

```
complex fto(-1:hsup,-ksup:ksup,-lsup:lsup)
write(file) a,b,c,alpha,beta,gamma,nort,hsup,ksup,lsup,sqhsup
write(file) fto
```

where:

<code>a,b,c,alpha,beta,gamma</code>	=	model-cell.
<code>nort</code>	=	orthogonalising code.
<code>hsup,ksup,lsup</code>	=	maximum indices of the array; the (h,k,l) indices run, respectively, within the limits (-1:hsup), (-ksup:ksup), (-lsup:lsup)
<code>sqhsup</code>	=	$1/\text{DMIN}^2$.
<code>fto</code>	=	array of complex molecular scattering factors.

In this case, it may be useful to provide also the `./o/tabl{#}.s` file, which is used by program OIC to define the integration radius and the model-cell for cross-rotation function calculations.

ROTING

ROTING calculates fast rotation functions (self-, cross- and locked-rotation functions). The rotation function (Patterson Overlap) is defined by

$$PO(\mathbf{R}) = \frac{1}{v} \int_{\Omega} P^{(t)}(\mathbf{r}) P^{(s)}(\mathbf{R}^{-1}\mathbf{r}) d^3\mathbf{r}$$

where $P^{(t)}$ and $P^{(s)}$ are the target and search Patterson functions, respectively. Note that the rotation applied to the search model is \mathbf{R} . By expanding the Patterson functions in spherical harmonics, PO may be casted into the form

$$PO(\phi, \theta, \psi) = \sum_{\ell=LMIN}^{LSUP} \sum_{m,m'=-\ell}^{\ell} \left[\sum_{n=1}^N \overline{e_{\ell,m,n}^{(t)}} e_{\ell,m',n}^{(s)} \right] d_{m,m'}^{\ell}(\theta) e^{i(m\phi+m'\psi)}$$

with

$$e_{\ell,m,n} = \sqrt{12\pi(2(\ell+2n)-1)} \sum_{\mathbf{H}} \frac{|F_{\mathbf{H}}|^2}{V} Y_{\ell,m}(\mathbf{H}/H) \frac{j_{\ell+2n-1}(2\pi HR)}{2\pi HR}$$

Calling command: `./e/roting {INP} {#} {OUT}`

Input: `./i/{INP}.i1` = input described below.

`./f/xudi` = binary file of packed and sorted H,K,L,F^{obs}.

`./f/tabl{#}.d` = binary file of molecular scattering factors for model {#}.

Output: `./f/elm{#}` = binary file of spherical-harmonics expansion coefficients for the crystal.

`./f/elmn{#}` = binary file of spherical-harmonics expansion coefficients for model {#}.

`./o/{OUT}.s` = rotation function output.

To recover the rotation function map, erase in script `./e/roting`, the symbol “#” in the line “`#mv $AMOREF/fort.93 rotmap`”. `./rotmap` is a binary file with several registers:

```
real rotmap(np,nf,nt),theta(nt)
write(file) np,nf,nt,p,f,(theta(n),n=1,nt)
do n=1,nt
write(file) ((rotmap(ip,if,n),ip=1,np),if=1,nf)
enddo
```

where:

`np,nf,nt` = number of sampling points in each direction.
`p,f` = ψ and ϕ Euler cell dimensions (in degrees).
`(theta(n), n=1,nt)` = θ sampling set (in degrees).

Input: file ./i/{INP}.i1 (general form)

```
roting +*+*+*+*+*+*+*+*+*
3 0 1 0 :printing
gene ++++++
...
elmn ++++++
...
rota ++++++
...
```

Description

1) Keyword (format A7) = 'roting '.

2) Printing options.

3) MKEY (format A4)

Keyword defining mode:	'gene'	generates model structure factors in a suitable cell;
	'elmn'	calculates spherical- harmonics expansion coefficients for crystal or models;
	'rota'	calculates rotation functions.

The three modes may be used independently.

Different modes:

“gene” mode:

```
gene ++++++
11 92 :logical units
title : model fragment number 1
127.790 98.030 90.050 90.000 90.000 90.000
x,y,z * end
1
15.00 3.50
0
```

Description

- 1) MKEY (format A4) = 'gene'.
 - 2) LUN1 LUN2
Logical units (see script ./e/roting):
LUN1 = input binary file of molecular scattering factors (./f/tabl{#}).
LUN2 = output binary file of packed and sorted Fourier coefficients (as produced by SORTING).
 - 3) Title (format A80).
 - 4) Model cell for rotation function (default cell = model-box + resolution + integration-radius).
 - 5) Symmetry operations (lower case), finishing '* end' (usually P1).
 - 6) NORT
Code to define an orthogonal reference frame (usually 1).
 - 7) DMAX DMIN
Resolution limits. Generates data within this range (in Å).
 - 8) NBOD
Option to generate structure factors with molecules placed at desired positions (usually 0).
-

“elmn” mode:

```
elmn ++++++
10 91 :logical units
* D44HEL **
99.700 167.300 84.700 90.000 90.000 90.000
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95.0 0.0
15.00 3.50 0.00 42.40
```

Description

- 1) MKEY (format A4) = 'elmn'.
 - 2) LUN1 LUN2
Logical units (see script ./e/roting):
LUN1 = input binary file of packed and sorted Fourier coefficients
(as produced by SORTING or ROTING -mode "gene").
LUN2 = output binary file of spherical-harmonics expansion
coefficients (./f/elm_x{#} or ./f/elm_n{#}).
 - 3) Title (format A80).
 - 4) Cell.
 - 5) Symmetry operations (lower case), finishing '* end'.
 - 6) NORT
Code to define an orthogonal reference frame.
 - 7) PERC BADD
PERC = uses only the PERC % highest F^{obs}.
BADD = B-factor added to F^{obs} (e.g. -5 to sharpen data).
 - 8) DMAX DMIN RMIN RMAX
DMAX |
DMIN | = resolution limits.
RMIN
RMAX = integration radii (in Å).
-

“rota” mode:

rota ++++++
91 92 9 10 11 :logical units
* D44HEL **
cross
6 500
2.5
0 0
0.5 1000 0.1E-04

Description

- 1) MKEY (format A4) = 'rota'.
 - 2) LUN1 LUN2 LUN3 LUN4 LUN5
Logical units (see script ./e/roting):
 - LUN1 = input binary file of spherical-harmonics expansion coefficients for fixed crystal (usually ./f/elm_x{#}).
 - LUN2 = input binary file of spherical-harmonics expansion coefficients for rotating crystal (usually ./f/elm_n{#}).
 - LUN3 = rotation function output (./o/{OUT}.s).
 - LUN4 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
 - LUN5 = input binary file of molecular scattering factors (./f/ta_l{#}).
 - 3) Title (format A80).
 - 4) RKEY (format A5)
Keyword defining mode:

'cross'	computes cross-rotation function;
'self'	computes self rotation.
 - 5) LMIN LSUP
Expansions between LMIN and LSUP are used. The ℓ -expansion controls the angular resolution. Low order terms are governed by the crystal symmetry; excluding them may reduce the final peak heights, but make the rotation parameters more precise and make multiple solutions have more equal heights.
 - 6) STEP
Step size for ϕ , θ and ψ (in degrees).
 - 7) NRS1 NRS2
Used to produce several shifts prior to the rotation function calculation (usually 0 0). This allows locked-rotation function calculations.
 - 8) CUTR NPIC DELT
 - CUTR = cutoff in rotation function output; first selects all peaks above CUTR \times maximum-peak-height.
 - NPIC = maximum number of peaks to output of rotation function.
 - DELT = used in peak-search to avoid spurious peaks.
-

Example

Self Rotation:

The self rotation may be calculated by using the following input cards:

```
roting +*+*+*+*+*+*+*+*+*
3 0 1 0 :printing
elmn ++++++
10 91 :logical units
* D44HEL **
99.700 167.300 84.700 90.000 90.000 90.000
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95.0 0.0
15.00 3.50 0.00 42.40
rota ++++++
91 0 9 0 0 :logical units
* D44HEL ** self
self
6 500
2.5
0 0
0.2 1000 0.1E-04
```

If the expansion-coefficients already exist (`./f/elm{x}{#}`), then the following input may be used:

```

rotng +*+*+*+*+*+*+*+*+*
3  0  1  0  :printing
rota ++++++
91  0  9  0  0  :logical units
* D44HEL ** self
self
6  500
2.5
0  0
0.2  1000  0.1E-04

```

Locked Cross-Rotation:

The locked cross-rotation is the average of the values of the cross rotation at orientations related by the NCS elements \mathbf{S}_n , $n = 1, \dots, N$:

$$PO_L(\mathbf{R}) = \sum_{n=1}^N PO(\mathbf{S}_n \mathbf{R}) / N .$$

By redefining the target function, it can be computed as an ordinary cross rotation,

$$\begin{aligned}
 PO_L(\mathbf{R}) &= \sum_{n=1}^N \frac{1}{v} \int_{\Omega} P^{(t)}(\mathbf{r}) P^{(s)}(\mathbf{R}^{-1} \mathbf{S}_n^{-1} \mathbf{r}) d^3 \mathbf{r} / N \\
 &= \frac{1}{v} \int_{\Omega} \left(\sum_{n=1}^N P^{(t)}(\mathbf{S}_n \mathbf{r}) / N \right) P^{(s)}(\mathbf{R}^{-1} \mathbf{r}) d^3 \mathbf{r} ,
 \end{aligned}$$

with the target Patterson function substituted by the average over the NCS of the rotated target functions. Let ϕ , θ , ψ be Euler angles of a self-rotation peak; the parameters of the inverse rotation are $\phi' = 180 - \psi$, $\theta' = \theta$, $\psi' = 180 - \phi$. Then, the locked cross-rotation may be calculated by using the following input ($N = 2$):

roting +*+*+*+*+*+*+*+*+*
3 0 1 0 :printing
rota ++++++
91 92 9 0 11 :logical units
* D44HEL ** locked
cross
6 500
2.5
2 0
0. 0. 0.
 ϕ' θ' ψ'
0.5 1000 0.1E-04

TRAINING

TRAINING calculates fast translation functions. The Fourier coefficient of the search model, rotated by \mathbf{R} and placed at \mathbf{T} , is given by the expression:

$$\begin{aligned} F_{\mathbf{H}}^{cal}(\mathbf{T}) &= \sum_{g=1}^G \left(f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}\mathbf{O}_m) e^{2\pi i\mathbf{H}\mathbf{t}_g} \right) e^{2\pi i\mathbf{H}\mathbf{M}_g\mathbf{T}} \\ &= \sum_{g=1}^G u_g^m(\mathbf{H}) e^{2\pi i\mathbf{H}\mathbf{M}_g\mathbf{T}} \end{aligned}$$

and the corresponding intensity by

$$\begin{aligned} I_{\mathbf{H}}^{cal}(\mathbf{T}) &= \sum_{g,g'=1}^G f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}_m\mathbf{O}_m) \overline{f_m(\mathbf{H}\mathbf{M}_{g'}\mathbf{D}\mathbf{R}_m\mathbf{O}_m)} e^{2\pi i\mathbf{H}((\mathbf{M}_g-\mathbf{M}_{g'})\mathbf{T}+\mathbf{t}_g-\mathbf{t}_{g'})} \\ &= \sum_{g,g'=1}^G u_g^m(\mathbf{H}) \overline{u_{g'}^m(\mathbf{H})} e^{2\pi i\mathbf{H}(\mathbf{M}_g-\mathbf{M}_{g'})\mathbf{T}}. \end{aligned}$$

Several translation functions may be calculated (“centered” variables are denoted by $\overline{I_{\mathbf{H}}} = I_{\mathbf{H}} - \langle I_{\mathbf{H}} \rangle$):

- centered-overlap

$$\begin{aligned} CO(\mathbf{T}) &= \sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{obs}} \times \overline{I_{\mathbf{H}}^{cal}(\mathbf{T})} \\ &\propto \sum_{g,g'=1}^G \sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{obs}} \overline{u_g^m(\mathbf{H})} u_{g'}^m(\mathbf{H}) e^{-2\pi i\mathbf{H}(\mathbf{M}_g-\mathbf{M}_{g'})\mathbf{T}} \end{aligned}$$

- Harada-Lifchitz

$$HL(\mathbf{T}) = \left(\sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{obs}} \times \overline{I_{\mathbf{H}}^{cal}(\mathbf{T})} \right) / \sum_{\mathbf{H}} I_{\mathbf{H}}^{cal}(\mathbf{T})$$

- correlation-coefficient

$$CC(\mathbf{T}) = \left(\sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{obs}} \times \overline{I_{\mathbf{H}}^{cal}(\mathbf{T})} \right) / \sqrt{\left(\sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{obs}}^2 \right) \times \left(\sum_{\mathbf{H}} \overline{I_{\mathbf{H}}^{cal}(\mathbf{T})}^2 \right)}$$

- phased-translation

1. without “external” phases

$$PT(\mathbf{T}) = \sum_{g,g'=1}^G \sum_{\mathbf{H}} \left(|F_{\mathbf{H}}^{obs}/u_g^m(\mathbf{H})| + |F_{\mathbf{H}}^{obs}/u_{g'}^m(\mathbf{H})| - 2 \text{ SCAL} \right) \\ \times \frac{\overline{u_g^m(\mathbf{H})}}{u_{g'}^m(\mathbf{H})} e^{-2\pi i \mathbf{H}(\mathbf{M}_g - \mathbf{M}_{g'}) \mathbf{T}}$$

2. with “external” phases

$$PTF(\mathbf{T}) = \sum_{\mathbf{H}} |F_{\mathbf{H}}^{obs}| e^{i\phi_{\mathbf{H}}^{ext}} \times \overline{F_{\mathbf{H}}^{cal}(\mathbf{T})}$$

- many-body translation

1. phased-translation

$$PTN(\mathbf{T}) = \sum_{\mathbf{H}} \left(\left| \frac{F_{\mathbf{H}}^{obs}}{F_{\mathbf{H}}^{fix}} \right| - \text{SCAL} \right) F_{\mathbf{H}}^{fix} \times \overline{F_{\mathbf{H}}^{cal}(\mathbf{T})}$$

2. for all others, replace

$$F_{\mathbf{H}}^{cal}(\mathbf{T}) \rightarrow F_{\mathbf{H}}^{cal}(\mathbf{T}) + F_{\mathbf{H}}^{fix}$$

Calling command: ./e/traing {INP} {OUT}

Input: ./i/{INP}.i1 = input described below.

 ./f/xudi = binary file of packed and sorted H,K,L,F^{obs}.

 ./f/ta1{#} = binary files of molecular scattering factors; {#} = 1,NTYP.

Output: ./o/{OUT}.s = translation function output.

To recover the last fast translation function map, erase in script ./e/traing, the symbol “#” in line “#mv \$AMOREF/fort.94 tramap”. ./tramap is a binary file with several registers:

```
real tramap(nx,ny,nz)
write(file) nx,ny,nz,a,b,c,alpha,beta,gamma
  do iz=1,nz
write(file) ((tramap(ix,iy,iz),ix=1,nx),iy=1,ny)
  enddo
```

where:

nx,ny,nz	=	number of sampling points in each direction.
a,b,c,alpha,beta,gamma	=	cell parameters (in Å and degrees).

Input: file ./i/{INP}.il

```
traing +*+*+*+*+*+*+*+*+*
7  1  1  0  1  0  1  1  :printing
10  9  :logical units
* D44HEL **
99.700 167.300 84.700 90.000 90.000 90.000
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95.0  0.0
15.00  3.50  0.88  1.00
3 ++++++
c-o  0.0
0.50  30  1.e-4
> 1  102.3  25.7  12.8  0.0755  0.3081  0.4560  27.6  52.1  54.0  42.0
> 1  135.5  43.0  320.5  0.4613  0.2053  0.0096  38.6  48.1  41.1  37.0
# 2   73.6  68.9  346.3  0.0000  0.0000  0.0000   7.6
3 ++++++
c-o  0.0
0.50  30  1.e-4
> 1  102.3  25.7  12.8  0.0755  0.3081  0.4560  27.6  52.1  54.0  42.0
> 1  135.5  43.0  320.5  0.4613  0.2053  0.0096  38.6  48.1  41.1  37.0
# 2   74.8  61.6  309.9  0.0000  0.0000  0.0000   4.0
...
```

Description

- 1) Keyword (format A7) = 'traing'.
- 2) Printing options.
- 3) LUN1 LUN2
Logical units (see script ./e/traing):
LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
LUN2 = translation function output (./o/{OUT}.s).
- 4) Title (format A80).
- 5) Cell.
- 6) Symmetry operations (lower case), finishing '* end'.
- 7) NORT
Code to define an orthogonal reference frame.
- 8) PERC BADD
PERC = uses only the PERC % highest F^{obs}.
BADD = B-factor added to F^{obs} (e.g. -5 to sharpen data).
- 9) DMAX DMIN TMIN SHAR
DMAX,DMIN = data resolution limits.
TMIN = fast translation function resolution (default value is DMIN/4).
SHAR = Shannon rate (the greater SHAR, the finer the mesh).
Defines the crystal cell sampling as
$$NX \approx 2 \times a / (TMIN/SHAR)$$
$$NY \approx 2 \times b / (TMIN/SHAR)$$
$$NZ \approx 2 \times c / (TMIN/SHAR)$$

Then add this information which is extracted from ROTING, TRAINING or FITING outputs:

- 10) NBOD
Number of molecules (n-body mode).
- 11) TKEY SCAL (format A5,*)
- | | | |
|------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TKEY | = | 'c-o' computes centered-overlap;
'p-t' computes phased-translation ('p-t-f' when phases
are available);
'h-l' computes Harada-Lifchitz translation function;
'c-c' computes correlation-coefficient. |
| SCAL | = | for phased-translation, to scale the phasing-model subtraction. |
- 12) CUTT NPIC DELT
- | | | |
|------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CUTT | = | cutoff in fast translation function output; first selects all peaks above CUTT \times maximum-peak-height of 'c-o', 'p-t', 'h-l' or 'c-c' fast translation function. |
| NPIC | = | maximum number of peaks to output of translation function. The program computes correlations and R-factors. |
| DELT | = | used in peak-search to avoid spurious peaks. |
- 13) # $\mu_m \phi_m \theta_m \psi_m x_m y_m z_m$ (NBOD cards; $m = 1, \text{NBOD}$)
This is a formatted line. The format is decided by the user when installing the package. If NBOD = 1 search is done over the Cheshire cell. Otherwise last will be translated over whole cell while keeping the others fixed.
- | | | |
|--------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| μ_m | = | model identification number; also logical-unit identifier for molecular scattering factors corresponding to the m -th molecule. Usually, logical-unit = $\mu_m + 10$. |
| $\phi_m \theta_m \psi_m$ | = | Euler angles for m -th molecule. |
| $x_m y_m z_m$ | = | translations (fractionary) for m -th molecule. For the last one, read but not used. |

Repeat 10) — 13) for other positions and orientations.

FITING

FITING performs rigid-body refinement. Minimizes

$$\sum_{\mathbf{H}} \left(F_{\mathbf{H}}^{obs} - \frac{e^{+B|\mathbf{H}|^2}}{Scale} \left| \sum_m F_{\mathbf{H}}^{cal(m)}(\phi_m, \theta_m, \psi_m, x_m, y_m, z_m) \right| \right)^2$$

with respect to B , $Scale$ and rotation and translation parameters.

Calling command: `./e/fitting {INP} {OUT}`

Input: `./i/{INP}.i1` = input described below.

`./f/xudi` = binary file of packed and sorted H,K,L, F^{obs} .

`./f/tabl{#}` = binary files of molecular scattering factors; $\{#\} = 1, NTYP$.

Output: `./o/{OUT}.s` = fast rigid-body refinement output.

Input: file ./i/{INP}.il

```
fiting +*+*+*+*+*+*+*+*+*
7  1  1  0  1  0  1  1  :printing
10  9  :logical units
* D44HEL **
99.700 167.300 84.700 90.000 90.000 90.000
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95.0  0.0
15.00  3.50
inertia tensors  2
  * 1  140.3  100.2  78.1
  * 2   80.0   66.1  50.0
3 ++++++
0  1  1  1  1  1  1
1  20  0.04
# 1  102.3  25.7  12.8  0.0755  0.3081  0.4560  27.6  52.1  54.0  42.0
# 1  135.5  43.0  320.5  0.4613  0.2053  0.0096  38.6  48.1  41.1  37.0
# 2   74.8  61.6  309.9  0.2951  0.9939  0.8339  43.9  45.5  41.8  28.0
...
```

Description

- 1) Keyword (format A7) = 'fiting '.
- 2) Printing options.
- 3) LUN1 LUN2
Logical units (see script ./e/fiting):
LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
LUN2 = fast rigid-body refinement output (./o/{OUT}.s).
- 4) Title (format A80).
- 5) Cell.
- 6) Symmetry operations (lower case), finishing '* end'.
- 7) NORT
Code to define an orthogonal reference frame.
- 8) PERC BADD
PERC = uses only the PERC % highest F^{obs}.
BADD = B-factor added to F^{obs} (e.g. -5 to sharpen data).
- 9) DMAX DMIN
DMAX,DMIN = resolution limits.
- 10) NTYP (format *, after column 15)
Number of different models whose positions are to be refined.
- 11) * m Q_{xx}^m Q_{yy}^m Q_{zz}^m Q_{xy}^m Q_{xz}^m Q_{yz}^m (NTYP cards; $m = 1, \text{NTYP}$)
This is a formatted line. The format is decided by the user when installing the package.
 m = model identification number.
 Q_{ij}^m = principal moments of inertia for m -th model.

Then add this information which is extracted from TRAINING or FITING outputs (or ROTING for space group P1):

- 12) NBOD
Number of molecules (n-body mode).
- 13) Refinement flags (7 flags).
Flags for refinement of B-factor, Euler angles and translations.
= $\begin{cases} 0 & \text{(no refinement);} \\ 1 & \text{(refinement).} \end{cases}$
- 14) NCYC NITE RMSS
 NCYC = number of times the NBOD bodies are alternately refined.
 NITE = number of iterations in the least-squares procedure.
 RMSS = root-mean-square shift (in Å). Least-squares stops if the rms. correction to positions is less than RMSS.
- 15) # $\mu_m \phi_m \theta_m \psi_m x_m y_m z_m$ (NBOD cards; $m = 1, \text{NBOD}$)
 This is a formatted line. The format is decided by the user when installing the package. If NBOD > 1, last is refined first, while the others are kept fixed; then proceeds cyclically.
 μ_m = model identification number; also logical-unit identifier for molecular scattering factors corresponding to the m -th molecule. Usually, logical-unit = $\mu_m + 10$.
 $\phi_m \theta_m \psi_m$ = Euler angles for m -th molecule.
 $x_m y_m z_m$ = translations (fractionary) for m -th molecule.

Repeat 12) to 15) for other positions.

MR2IC

MR2IC reads the rotation and translation applied by TABLING to the input models, and the refined rotations and translations output from FITING, to work out the final rotations and translations parameters to apply to the initial models.

Calling command: `./e/mr2ic {INP} {OUT}`

Standard input: `./e/mr2ic`

Input: `./d/data.d` = ascii file of main **AMOR ϵ** input.

`./o/tabl{#}.s` = ascii file of TABLING outputs;
{#} = 1,NTYP.

`./o/{INP}.s` = ascii file of FITING output.

Output: `./o/{OUT}.s` = output positional parameters.

Standard Input: (file `./e/mr2ic`)

mr2ic 1

Description

- 1) OKEY NUMF (format A6,*)
OKEY = 'mr2ic '
NUMF = selects up to a maximum of NUMF positions to output
(in decreasing order of correlation).
-

OIC

OIC creates inputs to the main molecular replacement programs by selecting and combining potential solutions, which are sorted according to the values of the correlation coefficient.

Calling command:

`./e/oic {NOM} {#} {OUT}` ; 3 arguments.

`./e/oic {NOM} {IN1} {OUT}` ; 3 arguments.

`./e/oic {NOM} {IN1} {IN2} {OUT}` ; 4 arguments.

Five situations are possible according to **{NOM}** and number of arguments:

- for ROTING inputs, model dimensions are read from file `./o/tabl{#}.s` ; 3 arguments.
- for one-body TRAINING inputs, orientations to translate are read from file `./o/{IN1}.s` ; 3 arguments.
- for n-body TRAINING inputs, orientations are read from file `./o/{IN1}.s`, and the fixed position from file `./o/{IN2}.s`. If orientations are read from a n-body output, only those corresponding to the last body are taken into account; 4 arguments.
- for n-body TRAINING inputs, where the orientations and the fixed position are read from the same file `./o/{IN1}.s` ; 3 arguments.
- for FITING inputs, positions to refine are read from file `./o/{IN1}.s` ; 3 arguments.

Input: `./i/{NOM}.i2` = input described below.

`./d/data.d` = ascii file of main **AMOR ϵ** input.

`./o/tabl{#}.s` = ascii files of TABLING outputs; {#} = 1,NTYP.

`./o/{IN1}.s` = ascii files of ROTING, TRAINING or FITTING
`./o/{IN2}.s` = outputs.

Output: `./i/{OUT}.i1` = OIC output, i.e. input to main molecular replacement programs.

The OIC program needs the `./o/tabl{#}.s` file to define the integration radius and the model cell for cross-rotation function calculations. This file is not created if the molecular scattering factors of the search model electron density is not calculated with TABLING.

Three different input modes:

Input: file `./i/{NOM}.i2` (ROTING mode)

```
oic roting +*+*+*+*+*+*+*+*+*
* D44HEL **
model: 1
15.00 3.50
1.0
2 10
2.5
0.5 1000
```

Description

- 1) Keyword defining mode (format A10) = 'oic roting'.
 - 2) Title (format A80).
 - 3) MTYP (format *, after column 7)
Model-identification number.
 - 4) DMAX DMIN
Resolution limits (in Å).
 - 5) RATE
Defines the integration radius as $\text{RATE} \times \text{Molrad}$, where Molrad is the radius of the smallest sphere, with origin at CoM, containing the whole molecule.
 - 6) LMINs LMINf
several rotation functions are calculated, where the spherical-harmonics expansions begin with $\text{LMINs} \leq \ell \leq \text{LMINf}$; the ℓ -expansion controls the angular resolution.
 - 7) STEP
Step size for ϕ , θ and ψ (in degrees).
 - 8) CUTR NPIC
CUTR = cutoff in rotation function output; first selects all peaks above $\text{CUTR} \times \text{maximum-peak-height}$.
NPIC = maximum number of peaks to output of rotation function.
-

Input: file ./i/{NOM}.i2 (TRAINING mode)

```
oic traing +*+*+*+*+*+*+*+*+*+*
* D44HEL **
n-body-s 1
p-t 1.0
0.0 10
15.00 3.50
0.5 30
fuzz: 2.0
over: 20.0
```

Description

- 1) Keyword defining mode (format A10) = 'oic traing'.
- 2) Title (format A80).
- 3) BKEY RANK (format A10,*)

BKEY	=	keyword defining mode. Three possibilities:	<div style="display: inline-block; vertical-align: middle; text-align: right;">'one-body '</div> <div style="display: inline-block; vertical-align: middle; text-align: right;">'n-body '</div> <div style="display: inline-block; vertical-align: middle; text-align: right;">'n-body-s '</div>
------	---	---------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The last one is for the particular case of n-body translations where the orientations to translate and the fixed position are read from the same file. When the number of arguments is 4, the OIC procedure uses “sed” to erase '-s'.

RANK = takes as fixed position the one that appears in RANK-th order of decreasing correlation (for n-body).
- 4) TKEY SCAL (format A5,*)

TKEY	=	<div style="display: inline-block; vertical-align: middle; text-align: right;">'c-o' computes centered-overlap;</div> <div style="display: inline-block; vertical-align: middle; text-align: right;">'p-t' computes phased-translation ('p-t-f' when phases are available);</div> <div style="display: inline-block; vertical-align: middle; text-align: right;">'h-l' computes Harada-Lifchitz translation function;</div> <div style="display: inline-block; vertical-align: middle; text-align: right;">'c-c' computes correlation-coefficient.</div>	
SCAL	=	for phased-translation, to scale the phasing-model subtraction.	
- 5) CORR NUMR

CORR	=	cutoff to select only those solutions with correlation coefficient greater than CORR.
NUMR	=	selects up to a maximum of NUMR orientations for input to translation (in decreasing order of correlation).
- 6) DMAX DMIN
Resolution limits (in Å).
- 7) CUTT NPIC

CUTT	=	cutoff in fast translation function output; first selects all peaks above CUTT × maximum-peak-height of 'c-o', 'p-t', 'h-l' or 'c-c' fast translation function.
NPIC	=	number of peaks to output of translation function. The program computes correlations and R-factors.
- 8) CUTO (format *, after column 5)
Cutoff to eliminate orientations to translate differing by less than CUTO degrees.

Description

- 1) Keyword defining mode (format A10) = 'oic fitting'.
 - 2) Title (format A80).
 - 3) CORR NUMT
CORR = cutoff to select only those solutions with correlation coefficient greater than CORR.
NUMT = selects up to a maximum of NUMT positions to refine (in decreasing order of correlation).
 - 4) DMAX DMIN
Resolution limits (in Å).
 - 5) NCYC NITE RMSR
NCYC = number of times the n-bodies are alternately refined.
NITE = number of iterations in the least-squares procedure.
RMSR = least-squares stops if the rms. correction to positions is less than RMSR \times DMIN.
 - 6) CUTO (format *, after column 5)
Cutoff to eliminate orientations differing by less than CUTO degrees (for space group P1).
 - 7) CUTD (format *, after column 5)
Cutoff to eliminate positions with CoM-CoM distance less than CUTD Å.
-

The OIC_ROTING, OIC_TRAINING and OIC_FITTING procedures

oic_rotting, oic_training and oic_fitting are the concatenations of OIC with the main molecular replacement programs.

OIC_ROTING

oic_rotting computes cross-rotation function.

Calling command: `./e/oic_rotting {NOM} {#} {OUT}`

equivalent to $\left| \begin{array}{llll} ./e/oic & \{NOM\} & \{#\} & \text{rot} \\ ./e/rotting & \text{rot} & \{#\} & \{OUT\} \end{array} \right.$

Description

`./i/{NOM}.i2` = input to OIC, ROTING mode.

`{#}` = model number.

`./i/rot.i1` = input to ROTING program.

`./o/{OUT}.s` = cross-rotation function output.

OIC_TRAINING

oic_training computes fast translation functions.

Calling command: `./e/oic_traing {NOM} {ANG} {OUT}`

equivalent to $\left| \begin{array}{l} ./e/oic \quad \{NOM\} \quad \{ANG\} \quad tra \\ ./e/traing \quad tra \quad \{OUT\} \end{array} \right.$

Description

`./i/{NOM}.i2` = input to OIC, TRAINING mode.

`./o/{ANG}.s` = filename of ROTING, TRAINING or FITING output; orientations to translate (and fixed positions in n-body cases) are read from this file. If the orientations are read from a n-body output, only those corresponding to the last body are taken into account.

`./i/tra.i1` = input to TRAINING program.

`./o/{OUT}.s` = translation function output.

Calling command: `./e/oic_traing {NOM} {ANG} {POS} {OUT}`

equivalent to $\left| \begin{array}{l} ./e/oic \quad \{NOM\} \quad \{ANG\} \quad \{POS\} \quad tra \\ ./e/traing \quad tra \quad \{OUT\} \end{array} \right.$

Description

`./i/{NOM}.i2` = input to OIC, TRAINING mode.

`./o/{ANG}.s` = filename of ROTING, TRAINING or FITING output; orientations to translate (and fixed positions in n-body cases) are read from this file. If the orientations are read from a n-body output, only those corresponding to the last body are taken into account.

`./o/{POS}.s` = ($\{POS\} \neq \{ANG\}$) filename of TRAINING or FITING output (or ROTING for space group P1); fixed positions are read from this file.

`./i/tra.i1` = input to TRAINING program.

`./o/{OUT}.s` = translation function output.

OIC_FITING

`oic_fiting` performs fast rigid-body refinements.

Calling command: `./e/oic_fiting {NOM} {POS} {OUT}`

equivalent to $\left| \begin{array}{lll} ./e/oic & \{NOM\} & \{POS\} \text{ fit} \\ ./e/fiting & \text{fit} & \{OUT\} \end{array} \right.$

Description

<code>./i/{NOM}.i2</code>	=	input to OIC, FITING mode.
<code>./o/{POS}.s</code>	=	filename of TRAINING or FITING output (or ROTING for space group P1); positions to be refined are read from this file.
<code>./i/fit.i1</code>	=	input to FITING program.
<code>./o/{OUT}.s</code>	=	fast rigid-body refinement output.
